

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA MATEMATIKO IN FIZIKO

Katja Horvat

**Računalniško krmiljenje
fotostimulatorja v povratni zanki z
odgovorom fotoreceptorjev**

DIPLOMSKO DELO
UNIVERZITETNI INTERDISCIPLINARNI ŠTUDIJSKI
PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN
MATEMATIKA

MENTOR: prof. dr. Igor Kononenko
SOMENTOR: doc. dr. Gregor Belušič

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko ter Fakulteta za matematiko in fiziko izdaja naslednjo nalogo:

Tematika naloge:

Z metodo elektroretinografijo, z znotrajceličnimi meritvami in z optičnimi metodami se meri delovanje mrežnice pri žuželkah. Za meritve je potreben zapleten fotostimulator, ki je sestavljen iz kompleta monokromatorja in nastavljivega absorpcijskega filtra. Oboje se uporablja za natančno nastavljanje intenzitete in barve dražljaja. Namen naloge je priprava programa za krmiljenje optičnih komponent in analizo elektrofizioloških signalov. Krmiljenje je omogočeno s platformo Arduino. Prvi korak je vzpostavljanje povratne zanke med meritvijo napetosti v celicah v očesu žuželke in sivinskim klinom ali ploščico z različnimi valovnimi dolžinami. Z mikroelektrodami se meri električni odgovor fotoreceptorjev in nevronov višjega reda. Po zajemanju elektrofizioloških podatkov z živali sledi analiza. Program naj izlušči standardizirane parametre iz razmeroma kompleksnega, vendar ponovljivega signala. Na osnovi velikosti parametrov naj z optičnimi pripravami v povratni zanki nastavlja zelen odgovor mrežnice. Iz nastavitve povratne zanke bomo lahko z visoko hitrostjo odčitavali občutljivost posameznih kategorij na svetlobo občutljivih celic. Platformo Arduino se programira v jeziku C++. Pri tem naj se izdelata tudi uporabniški vmesnik, da bodo lahko izvajalci meritev izboljšali časovno učinkovitost in ponovljivost poskusov.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Katja Horvat, z vpisno številko **63080041**, sem avtorica diplomskega dela z naslovom:

Računalniško krmiljenje fotostimulatorja v povratni zanki z odgovorom fotoreceptorjev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom prof. dr. Igorja Kononenka in somentorstvom doc. dr. Gregorja Belušiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 3. september 2014

Podpis avtorja:

Docentu dr. Gregorju Belušiču se zahvaljujem za pomoč, ideje in vodenje skozi celoten proces izdelave diplomske naloge. Hvaležna sem za vso znanje, ki sem ga pridobila v času sodelovanja.

Zahvaljujem se profesorju dr. Igorju Kononenku za čas in ažurnost, zaradi česar mi je nalogo uspelo realizirati v zadanih okvirjih.

Zahvala gre tudi profesorju dr. Marku Kreftu za pomoč in predloge ter vsem članom Skupine za integrativno fiziologijo in fiziologijo živali na Biotehniški fakulteti, ki so doprinesli k delovnem in prijetnem vzdušju.

Iskrena hvala Jeleni in Danielu za finančno pomoč in vzgojo, zaradi česar mi je bil omogočen razvoj do te točke.

Posebna zahvala gre tudi Gregu za podporo, spodbudo in potreben osebni prostor, skozi celoten študij.

Diplomsko delo posvečam vsem, ki se na kreativen in svojevrsten način lotevajo znanosti in doprinašajo k razvoju.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Razlaga sistema	5
2.1	Sistem kontrole sivega klina	6
2.2	CED 1401	7
2.3	WinWCP	8
3	Implementacija	9
3.1	Strojna oprema	9
3.2	Elektronska zasnova	10
3.3	Programska oprema	15
3.4	Rešitev problema	16
4	Sklepne ugotovitve	27
	Literatura	29

Seznam uporabljenih kratic

kratica	angleško	slovensko
IR	infra red	infrardeča
UV	ultra violet	ultravijolična
OD	optical density	optična gostota
T	transmittance	prepustnost
LED	light emitting diode	svetleča dioda
DAC	digital to analog converter	pretvornik iz digitalnega v analogni signal
ADC	analog to digital converter	pretvornik iz analognega v digitalni signal

Povzetek

Fotostimulator je sistem za proizvodnjo definiranih svetlobnih dražljajev pri raziskavah vida. Sistem je sestavljen iz več avtonomnih komponent; žarnice, monokromatorja, sivega klina, polarizatorja in komponent, ki so zadolžene za prenos, pretvorbo in interpretacijo zajetih informacij.

Diplomska naloga obravnava izdelavo kontrolnega sistema sivega klina. Komponenta, ki nadzoruje intenziteto projicirane svetlobe, je nadalje razdeljena na 5 manjših, med seboj povezanih delov. Intenziteto projicirane svetlobe nastavlja sivi klin, ki je vpet na trak, ki ga poganja koračni motor. Koračni motor z vzbujanjem signalov krmili gonilnik koračnega motorja, ki informacije dobi iz mikrokrmilniške ploščice Arduino. V Arduinu se nahaja logika za nastavitev in nadzor sistema sivega klina.

Prvi del diplomske naloge opisuje zgradbo in delovanje fotostimulatorja. V drugem delu se diplomska naloga osredotoča na komponento za kontrolo sivega klina. Natančneje opisuje pristope in metode, s katerimi smo se lotili problemov in razvili rešitev.

Izdelana rešitev se bo uporabljala pri dejanskem raziskovanju vida žuželk na Biotehniški fakulteti.

Ključne besede: Arduino, povratna zanka, fotoreceptor, signal, sivi klin.

Abstract

Photostimulator is a system for the delivery of defined light stimuli for vision research. The system consists of several autonomous components; light bulb, monochromator, continuously variable neutral density filter, polarizer and the components, designed for transfer and interpretation of captured data.

The first part of the thesis deals with implementation of density filter control system. The component, designed for controlling the intensity of projected light, consists of 5 smaller, coherent parts. The neutral density filter, attached to a linear rail, which is driven by stepper, sets the intensity of projected light. The stepper is controlled by a stepper driver, which gets the data from the Arduino Due microcontroller board. All logic, for setting and controlling the density filter system, is contained within Arduino.

The first part of the thesis describes the composition and functioning of photostimulator. In the second part, the thesis focuses on the density filter control system. It specifically describes the approaches and methods, which have been used for solving problems and developing the solution.

The implemented solution will be used in actual insect vision research on Biotechnical faculty.

Keywords: Arduino, feedback loop, photoreceptor, signal, density filter.

Poglavje 1

Uvod

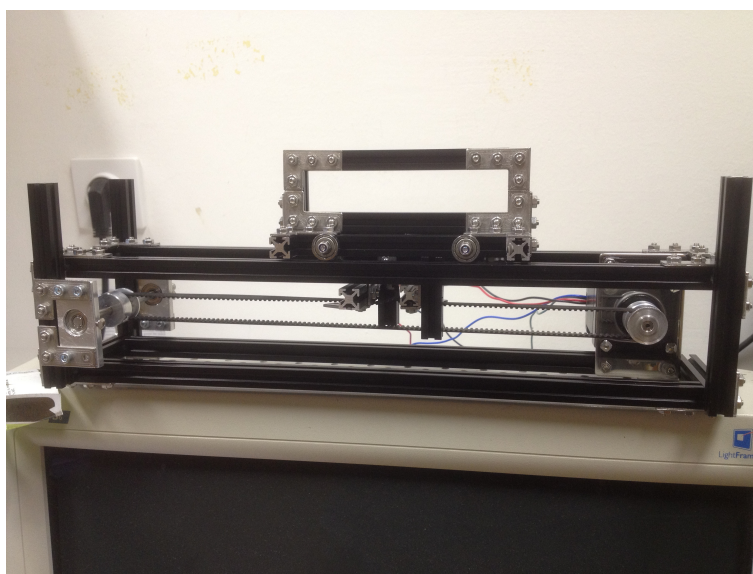
Vid je zelo pomemben čut za delovanje ljudi in živali. Pri človeku predstavlja večinski delež vseh sposobnosti za sprejemanje fizičnih dražljajev [3, 11]. Prav tako je pomembno zaznavanje svetlobnih dražljajev pri živalih. Na podlagi analize vida lahko povemo veliko o razvoju živali, hranjenju, razmnoževanju, . . . Na primer, če vemo kaj žival vidi in kaj jo privlači, lahko izdelamo vizualne vabe ter se tako izognemo ob določenih pogojih neefektivnemu in škodljivemu škropljenju pridelkov proti škodljivcem.

Za specifikacijo živalskega vida moramo natančno preučiti zgradbo in delovanje organa vida, očesa. To naredimo tako, da izmerimo fiziološke parametre, na primer električne aktivnosti nevronov in fotoreceptorjev. Da je postopek objektivni, kontroliran in ponovljiv, so na Biotehniški fakulteti, Oddelku za biologijo, Katedri za fiziologijo, antropologijo in etologijo, naredili večkomponentni sistem, fotostimulator, za meritev delovanja mrežnice pri žuželkah. Sistem projicira svetlobo nastavljive valovne dolžine, intenzitete in polarizacije v žuželče oko [8].

Diplomska naloga obravnava izdelavo sistema za krmiljenje optične komponente, sivega klina, ki nastavlja intenziteto projiciranega svetlobnega curka (sivi klin je prikazan na sliki 1.1). Komponenta za nastavitev intenzitete svetlobnega curka je že vgrajena v sistem in deluje, vendar je zaradi mehanske izvedbe sistema za krmiljenje sivega klina počasna in okorna. Zaradi načina implementacije uporabniku ni prijazna in omogoča izvajanje le osnovne funkcije, nastavljanje intenzitete svetlobnega curka na zeleno vrednost. Pojavila se je želja po pospešitvi sistema in možnosti prenosa kontrolnega dela sistema. Isto kontrolo si želijo upo-

rabiti tudi na Očesni kliniki, na Univerzitetnem kliničnem centru Ljubljana, za raziskovanje vida pri ljudeh. Odločili so se, da ga zaradi lažje mehanske izvedbe in cene zamenjajo z linearnim. Vključili smo še nekaj drugih komponent, med drugim komponento za samogradnjo 3D tiskalnikov in mikrokrmilniško ploščico Arduino, ki deluje kot kontrolna enota sistema. Z vgradnjo slednje smo sistem naredili kompakten in fizično prenosljiv, po drugi strani pa smo pridobili nadzor nad pozicijo sivega klina v dejanskem času. To nam je omogočilo implementacijo dodatne funkcije; krmiljenje fotostimulatorja v povratni zanki z odgovorom fotoreceptorja.

Povratna zanka je princip delovanja sistema, pri katerem sistem popravljamo v nastavljen, vhodno stanje na podlagi povratne informacije [14, 22]. Uporabniku smo želeli omogočiti možnost fiksiranja informacije odgovora na želeno vrednost. Sistem naj se v ozadju informaciji odgovora prilagaja povsem avtonomno. S tem uporabniku omogočimo pridobivanje novih podatkov. Uporabnik bo imel informacijo, za koliko je bilo potrebno popraviti intenziteto svetlobnega curka, da je stanje napetosti fotoreceptorja ostalo nespremenjeno. Poleg tega dodajanje novih faktorjev v poskus ne bo več vplivalo na električni odziv fotoreceptorja, s čimer se bo osamilo vpliv na druge objekte raziskave [15, 9].



Slika 1.1: Mehansko ogrodje za sivi klin. Klin je vgrajen v pravokotni okvir na vozičku na linearni tračnici. Koračni motor se nahaja na osi na desni strani slike.

Poglavje 2

Razlaga sistema

Na Biotehniški fakulteti, Oddelku za biologijo, Katedri za fiziologijo, antropologijo in etologijo, so za analizo žuželčjega očesa sestavili sistem, ki projicira svetlobni curek, kontroliranih karakteristik, v oko živali.

Prvi element fotostimulatorja je 150W XBO ksenonska žarnica (Osram, Nemčija), ki oddaja belo svetlobo. Svetlobni curek potuje skozi monokromator (B&M, Nemčija), optomehansko komponento, ki svetlobnemu curku nastavi valovno dolžino v intervalu od 300 do 700 nm. Fotoreceptorji očesa živali so fazično-tonične čutne celice, zato ob spremembi svetlobe močno reagirajo, potem pa se odziv zmanjša [4]. Svetlobni curek prekinja mehanska bliskovnica (Thorlabs, Nemčija)(ang. chopper), majhna jeklena vrtljiva vetrnica. Nadalje nastavljamo intenziteto svetlobnega curka s sivim klinom (NDL-10C-4, Thorlabs, Nemčija). Sivi klin je pravokotni trak iz kvarčnega stekla (kvarčno steklo se od ostalih stekel razlikuje po tem, da prepušča UV svetlobo [6]) z naporjenim reflektivnim, barvno nevtralnim, kovinskim nanosom [5], ki ustvarja gradient optične gostote od 0 do 4. Optično gostoto izračunamo po enačbi (2.2), kjer je I_0 intenziteta sevane svetlobe točkastega svetila in I_1 intenziteta svetlobe, ki je prešla skozi material.

$$OD = -\ln T \quad (2.1)$$

$$= -\ln \frac{I_1}{I_0}. \quad (2.2)$$

Svetlobnemu curku se na polarizatorju iz polarizacijske folije (OUV2500, Knight optical, Velika Britanija) nastavi še linearno polarizacijo svetlobe oz. smer

e-vektorja fotonov. Svetlobni curek ima na tem mestu točno definirane vse karakteristike, ki vplivajo na preučevane odzive fotoreceptorja v očesu živali.

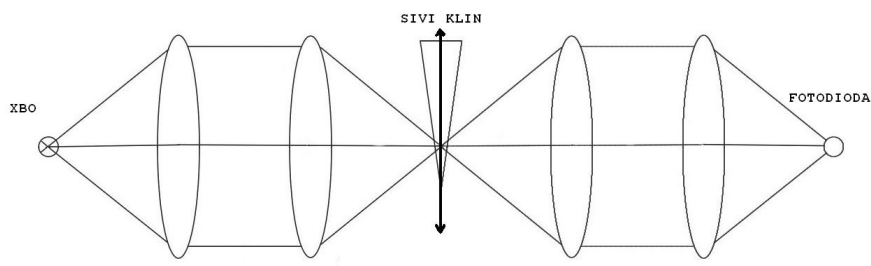
Za pridobivanje signalov iz posamezne fotoreceptorske celice, v njo vstavimo stekleno elektrodo. To je mikropipeta s konico premera 50-200 nm, napolnjena z (do 3 M) nasičeno raztopino elektrolita KCl. Raztopina je z Ag/AgCl elektrodo povezana z ojačevalnikom (SEC 10 LX, Npi, Nemčija). Ob osvetlitvi fotoreceptorja pride do spremembe električne napetosti na membrani celice, ki jo z elektronskim ojačevalnikom ojačamo do amplitude 0-3.3 V in digitaliziramo z Micro 1401 laboratorijskim vmesnikom (CED, Cambridge, Vel. Britanija, v nadaljevanju CED 1401). To je pretvornik iz analognega v digitalni signal, iz digitalnega signala v analogni, ter generator digitalnih TTL pulzov. CED 1401 posreduje ojačan signal na računalnik naloženemu programu WinWCP. Signal sestavlja časovni potek električne napetosti in ga razumemo kot električni odgovor fotoreceptorja.

2.1 Sistem kontrole sivega klina

Svetlobni curek se razseka z mehansko bliskovnico. Ta ima pritrjeno fotoelektrično stikalo, ki pošilja signal, zaradi katerega imamo v vsakem trenutku informacijo o tem ali je svetlobni curek prekinjen ali ne. Sivi klin je vstavljen v gorišče teleskopske konfiguracije s povečavo 1, ki jo tvorita dve plankonveksni kvarčni leči, kot je prikazano na sliki 2.1. S tem zagotavljamo natančno nastavitev jakosti svetlobnega curka, saj je svetloba na klinu zbrana v majhni točki. Sistem zaradi linearnosti deluje na preprostem principu koračnega pomikanja. Uporabnik nastavi želeno vrednost, ki se pretvori v korak, na katerega naj se sivi klin pomakne.

2.1.1 Sivi klin pred adaptacijo

Sivi klin je bil v obstoječem sistemu počasen in okoren. Uporabnik je v ukazni vrstici zagnal program in vanj vnašal tekstovne datoteke z vrednostmi, ki so sivemu klinu nastavljale pozicijo. Program, ki je bral vhodne datoteke, je bil implementiran na principu branja vrednosti iz tabele. Za vsako vrednost je že vnaprej definirano, na katero pozicijo naj se sivi klin pozicionira. Postopek je kompliciran za uporabnike, ki niso večji dela z ukaznimi vrsticami, po drugi strani pa ne deluje v dejanskem času, kar onemogoča implementacijo dodatnih funkcij. Zaradi



Slika 2.1: Sivi klin med ksenonsko žarnico in fotodiodo v 4f konfiguraciji leč

arhitekture mehanskega dela sistema se postopka ni dalo preprosto pospešiti in poenostaviti.

2.1.2 Linearni sivi klin

Krožni sivi klin smo zamenjali z linearnim in s tem pridobili na preprostosti mehanskega dela sistema. Na račun preprostosti nismo izgubili nobene karakteristike sivega klina, omogoča pa nam pospešitev in implementacijo dodatnih funkcij. Zaradi zamenjave sivega klina smo morali nadomestiti večino strojne opreme, ki premika sivi klin. Opisana je v razdelku 3.1. Sistemu smo dodali novo komponento, mikrokrmilniško ploščico, ki zaradi delovanja v dejanskem času omogoča realizacijo teh funkcij.

2.2 CED 1401

CED 1401 je laboratorijski vmesnik, pretvornik iz analognega v digitalni signal, s 16 kanali ADC, 4 kanali DAC, 5 kanali za merjenje časa in 6 kanali (D/A, TTL) z urami. Ima svoj procesor in spomin. Signal pripeljemo na zelen vhod, s katerega v računalniškem programu beremo digitalizirano vrednost. Obratno iz računalniškega programa nanj pošljemo vrednost in na izhodu beremo analogen signal.

2.3 WinWCP

WinWCP je odprtokodni program, izdelan za potrebe na področju elektrofiziologije, ki ga je razvil John Dempster na Univerzi Strathclyde, v Veliki Britaniji. Za delovanje potrebuje laboratorijski vmesnik (v našem primeru CED 1401), preko katerega prejeme signale ali pa nanj pošilja vrednosti. Je DAC, ter generator digitalnih TTL pulzov, vsebuje prijazen grafični vmesnik, ki omogoča izrisovanje in analiziranje signalov, nastavitev protokolov... Omogoča tudi kreiranje poljubnih analognih in digitalnih signalov, ki jih pošilja preko laboratorijskih vmesnikov. Primeri prikaza signalov so na sliki 3.9, 3.7, 3.8 in 3.6.

Poglavje 3

Implementacija

3.1 Strojna oprema

3.1.1 Koračni motor

Koračni motor je komponenta, ki z vzbujanjem določa zasuk osi za določeno razdelbo kroga. Zasuk motorja se prek mehanizma prenese na pogon, ki definira korak na traku, kar pomeni dejanski premik vozička, pripetega na trak. Koračni motor, vgrajen v naš sistem (Mercury SM-42BYG011-25, LR Kitajska), ima definiranih 200 korakov na krog, to je 1.8° na korak.

3.1.2 Gonilnik koračnega motorja

Gonilnik koračnega motorja (SMSD-4.2, Smart motor devices, Estonija) je naprava, ki določa rotacijo osi koračnega motorja. Ima dva digitalna TTL vhoda, ki sprejemata signale za proženje koraka in določitev smeri koračnega motorja. Vsebuje mehanska stikala, s katerimi določamo, koliko premikov osi pomeni en korak. To nam določa natančnost na račun hitrosti. Dolžino koraka smo določili empirično, da smo dosegli optimalno razmerje med hitrostjo in natančnostjo.

3.1.3 Arduino Due

Arduino Due (Arduino, Italija) je mikrokrmilniška ploščica, ki temelji na 32-bitnem ARM mikrokontrolerju Atmel. Ima 54 digitalnih vhodov/izhodov, 12 analognih

izhodov, 2 kanala, ki pretvarjata iz digitalnega v analogni signal, uro, ki deluje na 84 MHz, USB povezavo zmogljivosti OTG, gumb za resetiranje in gumb za izbris podatkov iz ploščice in več drugih komponent, ki za razvoj našega sistema niso pomembne. Za razliko od ostalih Arduino ploščic Arduino DUE deluje na napetosti 3.3 V. Povezavo z računalnikom smo vzpostavili preko USB vrat. Ploščica vsebuje vse, kar potrebujemo za delo z mikrokontrolerjem [13, 20].

3.1.4 Mehanska bliskovnica

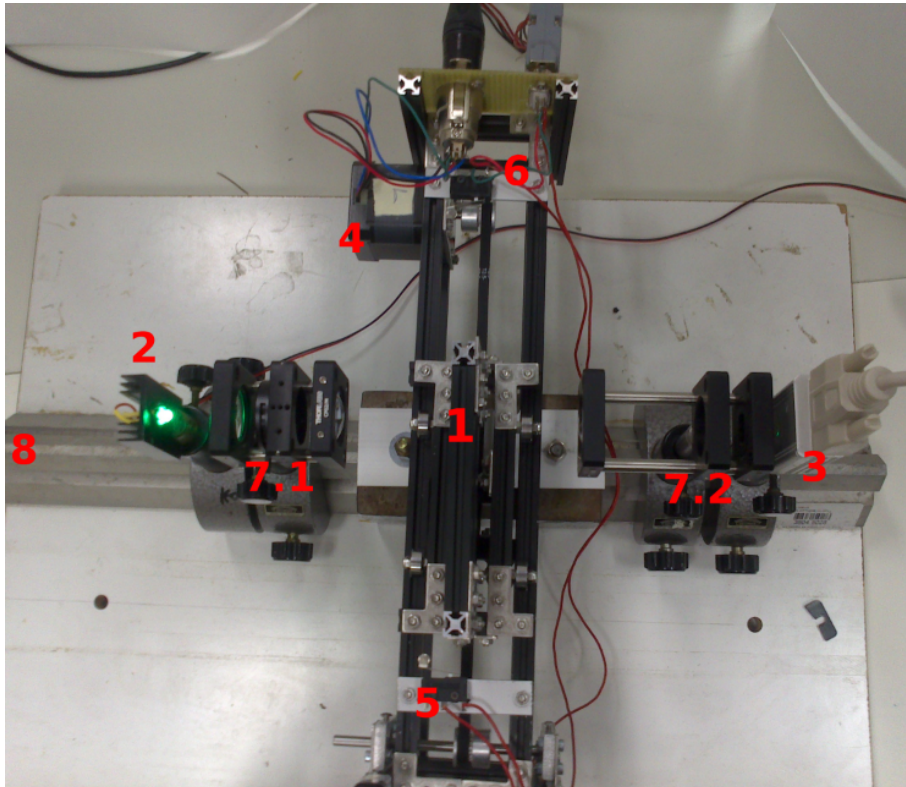
Mehanska bliskovnica je jeklena vetrnica, ki se vrti s konstantno, vendar nastavljivo hitrostjo. Ko jo namestimo v smeri svetlobnega curka, ga lopatice prekinjajo. Na statičnem stojalu je na nasprotni strani nameščeno fotoelektrično stikalo, sestavljeno iz IR LED in fotocelice. Ko projiciran svetlobni curek zadane katero od lopatic, hkrati druga lopatica prekine tudi IR svetlobni curek LED, kar zazna fotocelica na nasprotni strani. S tem se spremeni stanje signala, kateri nam v vsakem trenutku daje informacijo o tem ali je svetlobni curek prekinjen ali ne.

3.2 Elektronska zasnova

Z mikrokrmilniško ploščico krmilimo gonilnik koračnega motorja. Pošiljamo mu digitalne vrednosti za korak in smer. Kadar želimo narediti en korak, mu preko določene nožice (ang. pin) pošljemo signal z napetostjo 3.3 V, naredimo minimalni premor (npr. 1 ms) in pošljemo signal z napetostjo 0 V. Smer definira signal, ki je bil poslan iz druge nožice. Sistem smo postavili in definirali tako, da signal napetosti 3.3 V predstavlja desno smer, ko pa signalu napetost nastavimo na 0 V, določimo levo smer. Definicijo sistema prikazuje slika 3.1.

Konstrukcija, na katero sta pritrjena koračni motor in trak, ima na obeh kritičnih točkah pritrjeno stikalo, ki deluje na napetosti 3.3 V. Kadar je stikalo sklenjeno mikrokrmilniški ploščici pošlje digitalni signal napetosti 3.3 V. To programu daje podatek o skrajni poziciji vozička, vpetega na konstrukcijo.

Mikrokrmilniška ploščica dobiva informacije na dveh digitalnih (informacija o skrajni legi vozička) in treh analognih vhidih. Vrednost dobljeno na analognem vhodu mikrokrmilniška ploščica pretvori v niz bitov. Napetost, ki jo pošiljamo na ploščico je lahko znotraj intervala $[0, 3.3 \text{ V}]$. Dobljene vrednosti na analognem



Slika 3.1: Mehanski del prototipa sistema sivega klina v razvojni fazi iz ptičje perspektive. 1: sivi klin; 2: LED; 3: fotodioda; 4: koračni motor; 5: stikalo, ki določa kritično točko na desni strani sistema; 6: stikalo, ki določa kritično točko na levi strani sistema; 7.1 in 7.2: skupaj 4 plankonveksne kvarčne leče; 8: aluminijasto vodilo, na katero je pritrjena konstrukcija

vhodu zaradi digitalizacije mikrokrmilniške ploščice dobimo v intervalu $[0, 1023]$, za privzeto 10 bitno natančnost oz. vrednost v intervalu $[0, 4095]$, za 12 bitno natančnost, ki nam jo omogoča zmogljivejša ploščica Arduino Due [20]. Na prvem vhodu prejme vrednost, ki jo je uporabnik nastavil v programu WinWCP. Vrednost preberemo v 12 bitni natančnosti in jo pretvorimo iz intervala vhodnih bitov $[0, 2^{12}]$ v interval korakov $[0, \text{steviloKorakov}]$ po enačbi (3.1). Spremenljivke v enačbi predstavljajo vhodni interval $[\text{inMin}, \text{inMax}]$, izhodni interval $[\text{outMin}, \text{outMax}]$, vrednost, ki jo pretvarjamo x in izračunana vrednost $value$.

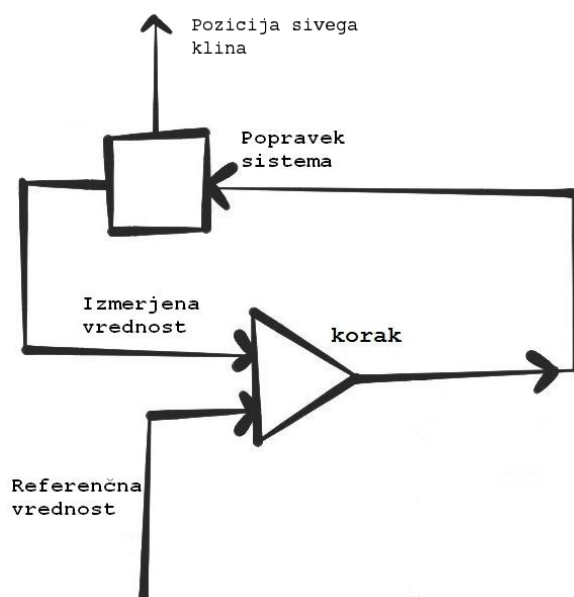
$$value = (x - \text{inMin}) * \frac{\text{outMax} - \text{outMin}}{\text{inMax} - \text{inMin}} + \text{outMin} \quad (3.1)$$

Vrednost signala stikala, ki je nameščen na sekalcu svetlobe, se prejme na digitalnem vhodu. Digitalna vrednost, ki jo dobimo je lahko v stanju visoke napetosti (*HIGH*), kar pomeni, da je stikalo sklenjeno ali v stanju nizke napetosti (*LOW*), kar pomeni, da je stikalo prekinjeno. To nam daje informacijo o tem ali je projiciran svetlobni curek v danem trenutku prekinjen.

Za delovanje povratne zanke je ključnega pomena vrednost, ki jo dobimo na 3. analognem vhodu. Dobimo jo iz fotodiode, pretvornika, ki na podlagi fotoelekta svetlobo pretvori v električni tok [21, 2] in za čas razvoja simulira fotoreceptor žuželčjega očesa. Tok iz fotodiode smo pretvorili v električno napetost s pomočjo logaritemskega ojačevalnika (ADL5310-Eval, Analog Devices, ZDA) in jo posredovali laboratorijskemu vmesniku CED 1401, ki razpošlje signal v računalnik, program WinWCP, na eni strani in v mikrokrmilniško ploščico Arduino na drugi. Signal bi v razvojni fazi teoretično moral biti pravilne kvadratne oblike, vendar prihaja do odstopanj zaradi napak, ki so posledica meritev, ojačitve signala in prenosa podatkov. Zgornja in spodnja amplitudna vrednost signala sta v razvojni fazi usklajeni z visokim in nizkim stanjem napetosti. V primeru dejanske uporabe, fotodiodo zamenja fotoreceptor žuželčjega očesa, ki ima reakcijsko latenco tipično 20 ms in nekvadratno, fazično-tonično obliko [4].

3.2.1 Povratna zanka

Povratna zanka je nadzorni del programa, naprave, sistema, ki omogoča popravek odstopanja od nastavljene vrednosti, na podlagi primerjave le te in vhodnega signala [22, 14]. Shema načina delovanja povratne zanke je prikazana na sliki 3.2.



Slika 3.2: Shema delovanja povratne zanke

Točko nastavitve v našem sistemu predstavlja zelena napetost na membrani fotoreceptorja. To je vrednost, ki jo nastavi uporabnik. Vhodno stanje predstavlja izmerjena napetost odziva fotoreceptorja [9]. Signal iz fotoreceptorja je kontaminiran z električnim in biološkim šumom, je fazično-tonične oblike ter ga v nadaljevanju obravnavamo kot funkcijo. Zaradi adaptacijskih sposobnosti fotoreceptorja, opisanih v razdelku 2 in počasnega drsenja napetosti, ki je posledica neidealnih pogojev meritve, konstantno prekinjamo svetlobni curek in s tem signal naredimo periodičen [4, 7]. Funkcijo odziva fotoreceptorja s povprečenjem vrednosti signala znotraj ene periode poenostavimo na kvadraten signal in s tem prihranimo nekaj časa na račun natančnosti, ki na tem mestu ni pomembna. Fotoreceptorji najsposobnejših lovcev med žuželkami, kačjih pastirjev, imajo razmerje med signalom in šumom tako neugodno, da svetlobnega kontrasta manjšega od 1 % niso sposobni zaznati [16, 12, 1].

Dokaz. Zveza med korakom na traku in optično gostoto je linearna, zato korak x pretvorimo po enačbi (3.1) iz intervala $[0, 780]$ v interval $[0, 4]$.

$$OD = \frac{4 * x}{880} = \frac{x}{220} \quad (3.2)$$

Izračunana vrednost OD predstavlja optično gostoto oz. absorbanco materiala. Iz enačbe (2.1) izrazimo prepustnost T

$$T = 10^{-OD} \quad (3.3)$$

Prepustnost po zvezi iz enačb (2.1) in (2.2) pomeni razmerje med intenziteto svetlobe, ki je prešla material in intenziteto sevane svetlobe in predstavlja svetlobni kontrast.

$$T = \frac{I_1}{I_0} \quad (3.4)$$

Funkcija, ki preslika iz intervala korakov v interval intenzitete je logaritemska, z zalogo vrednosti v $[0.0001, 1]$. Dokazati moramo, da se intenziteta pri spremembi za 1 korak ne spremeni za več kot 1 %, na območju najhitreje rastoče funkcije.

$$|T_0 - T_1| = |10^{-\frac{x_0}{220}} - 10^{-\frac{x_1}{220}}| \quad (3.5)$$

$$= |10^{-\frac{0}{220}} - 10^{-\frac{1}{220}}| \quad (3.6)$$

$$= |1 - 0.98959| \quad (3.7)$$

$$= 0.01041 \quad (3.8)$$

Intenziteta se ob premiku sivega klina za en korak na najbolj kritični točki spremembe kontrasta spremeni za približno 1.041 %. To je le malo nad mejo zaznave kontrasta najsposobnejšega žuželčjega fotoreceptorja in je za naš sistem zadovoljivo. \square

Na izhodu komparatorja nastavljene vrednosti in vhodnega signala dobimo število korakov, ki predstavlja korekcijo sistema, da se bo vhodno stanje ponovno ujemalo s točko nastavitve.

Izhod povratne zanke je trenutna pozicija sivega klina, ki smo jo definirali s številom korakov [15, 7, 9]. Korak pošljemo preko laboratorijskega vmesnika CED 1401, v računalniški program WinWCP v obliki analognega signala v arbitrarno določenem intervalu $[0, 3 \text{ V}]$. Uporabniku se v dejanskem času izrisuje signal, ki predstavlja pozicijo sivega klina.

3.3 Programska oprema

3.3.1 Arduino IDE

Za razvoj mikrokrmilniške ploščice smo uporabljali odprtokodno okolje Arduino IDE 1.5.7. Okolje je sicer še v testni verziji beta, vendar je edino kompatibilno z mikrokrmilniško ploščico Arduino Due [19, 20]. Vsebuje tekstovni urejevalnik za pisanje kode, sporočilno polje, tekstovni upravljalnik, orodno vrstico s pogostejšimi funkcijami in več menijev. Poveže se z mikrokrmilniško ploščico, na njo naloži program, v določenih primerih pa z njo tudi komunicira. Podpira programski jezik Arduino, ki je v bistvu nabor C/C++ funkcij, ki jih lahko uporabljamo v našem programu. Programsko kodo Arduinov prevajalnik prevede tako, da naredi nekaj manjših popravkov, potem pa jo pošlje direktno prevajalniku C/C++ (avr-g++).

V tekstovnem urejevalniku pišemo skice (ang. sketch). Sestavljata jo dve posebni funkciji *setup* in *loop*. Funkcija *setup* se zažene samo enkrat, ob zagonu skice in jo po navadi uporabimo za inicializacijo knjižnic, stanj nožic, spremenljivk... Funkcija *loop* pa se izvaja znova in znova in ponavadi predstavlja jedro programa. Program mora vedno vsebovati ti dve funkciji, tudi, če ju ne potrebujemo [13]. Tekom razvoja je bilo ugotovljeno, da mikrokrmilniška ploščica najverjetneje ne podpira večnitnega programiranja.

3.3.2 Processing

Program, napisan v programskem jeziku Arduino, naložimo na mikrokrmilniško ploščico, ki od tega trenutka dalje deluje avtonomno in nima vzpostavljene povezave z računalnikom. To zelo otežuje razhroščevanje, brez česar bi bil razvoj programa, kot ga narekuje diplomska naloga, pretežak. Programski jezik Arduino, ki je sicer razvit na osnovi Processing, ne omogoča uporabe vseh knjižnic, zato smo za spremljanje poteka programa in vrednosti spremenljivk uporabili program Processing. To je odprtokodni program z integriranim razvojnim okoljem.

3.4 Rešitev problema

3.4.1 Teoretično

Nastavitev pozicije sivega klina na fiksno vrednost

Uporabnik na računalniku, v programu WinWcp, nastavi želeno vrednost. Program omogoča nastavljanje vrednosti električne napetosti, v enoti milivolt, arbitrarno določene ciljne amplitude signala iz fotodiode. Arduino Due sprejema vrednosti od 0 do 3.3 V, prekoračitev te vrednosti pa lahko poškoduje mikrokrmilniško ploščico, zato naj uporabnik vnese vrednost v intervalu [0, 3300 mV].

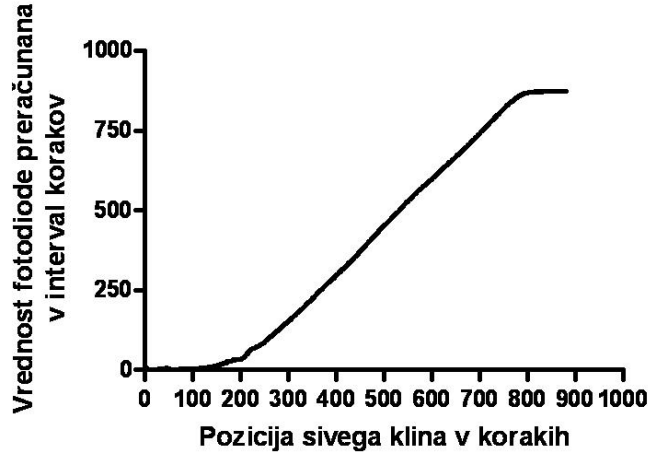
Za implementacijo sistema smo izbrali linearni sivi klin, zato je zveza med vrednostjo električne napetosti in intenziteto svetlobe linearna. Med razvojem se je pojavil problem, ko zveza ni bila linearna, zaradi fizične nastavitve skrajne lege traku. Iz slike 3.3 je jasno razvidno, da se intenziteta svetlobe prvih 200 in zadnjih 100 korakov skoraj ne spreminja. Signal je na začetku zelo šumen in nestabilen, saj pri tako majhnem gradientu optične gostote fotodiode, kot tudi marsikatero oko, ni sposobno natančno zaznati spremembe [2]. Zadnjih 100 korakov je steklo brez reflektivnega nanosa in prav tako sprememb v zaznavanju intenzitete svetlobe ni. Odločili smo se, da pomaknemo skrajno levo točko za približno 200 korakov v desno, skrajno desno točko pa za 100 korakov v levo. Zveza je zdaj linearna, vseeno pa vsebuje majhen šum, tipično 1 korak, kar smo že dokazali, da ne vpliva na natančnost našega sistema, saj tipičen žuželčji fotoreceptor ni sposoben zaznave kontrasta manjšega od 1 % [16, 12, 1].

Mikrokrmilniška ploščica v tej fazi implementacije še nima povratnih informacij fotodiode oz. fotoreceptorja, ima pa podatke o lokaciji vozička vpetega na trak, torej podatke o poziciji sivega klina. Analogno vrednost preračunamo iz intervala bitov v interval korakov (razlaga v podrazdelku 3.2). Dobljeno vrednost primerjamo s korakom, na katerem je trenutno voziček in ga po potrebi premaknemo na novo pozicijo.

Nastavitev pozicije sivega klina v povratni zanki

Povratna zanka deluje na principu prilagajanja povratni informaciji, v našem primeru kvadratnemu signalu ene periode (razloženo v razdelku 3.2.1), pridoblje-

Nelinearna zveza med pozicijo sivega klina in vrednostjo iz fotodiode



Slika 3.3: Slika prikazuje nelinearno zvezo med električno napetostjo, dobljeno iz fotodiode, preračunano v interval korakov in korakom koračnega motorja

nemu iz fotoreceptorja oz. fotodiode v razvojni fazi. Funkcijo signala pretvorimo v konkretno vrednost z metodo pulzne modulacije. Pulzna modulacija je tehnika prilagajanja, ki kontrolira dolžino pulza na aproksimirani informaciji signala. Uporablja kvadratno obliko signala, katerega dolžina aproksimira povprečno vrednost signala. Funkcijo signala $f(t)$ poenostavimo na visoko vrednost signala y_{max} , za $0 < t < D \cdot T$ in nizko vrednost signala y_{min} , za $D \cdot T < t < T$, pri čemer je T čas ene periode, D obratovalni cikel, $D \cdot T$ pa predstavlja dolžino signala. Povprečno vrednost izračunamo po enačbi (3.9) [18].

$$y = \frac{1}{T} \left(\int_0^{DT} y_{max} dt + \int_{DT}^T y_{min} dt \right) = D \cdot y_{max} + (1 - D) y_{min}. \quad (3.9)$$

Izraz obratovalni cikel D opisuje razmerje med časom, ko je napetost visoka in celotnim intervalom oz. periodo. Izraža se z odstotkom; 100 % pomeni najvišjo vrednost napetosti [18].

Vrednosti integrala v skrajnih legah nam določata interval, iz katerega preslikamo dobljeno vrednost v interval korakov, po enačbi (3.1).

Potek programa: ko uporabnik vnese vrednost, na katero želi nastaviti odziv fotoreceptorja, pomaknemo voziček na preračunan korak na traku. Vrednost nam bo služila kot referenčna vrednost. Nato ves čas beremo vrednosti signala, ki jih dobivamo iz fotodiode oz. fotoreceptorja, jih povprečimo, da dobimo kvadratno obliko signala in po enačbi (3.9) pretvorimo v povprečno vrednost signala. Vrednost po enačbi (3.1) pretvorimo v interval korakov in podobno, kot v razdelku 3.4.1, med seboj primerjamo vrednosti. Ob potrebi premaknemo voziček na potrebno pozicijo. Med pomikanjem na vsakem koraku uporabniku vračamo informacijo o trenutni legi vozička.

3.4.2 Praktično

Program je definiran tako, da se najprej izvede funkcija *setup* (opisano v razdelku 3.3.1). V njej definiramo stanja nožic, nato pa izvedemo klic funkcije *Init*, kjer sistem skalibriramo. Kalibracija, ki se izvede ob vsakem zagonu, sistem naredi neodvisen in prilagodljiv. Kalibracijo izvedemo tako, da pomikamo voziček z vpetim sivim klinom od skrajne leve lege, do skrajne desne in hkrati štejemo korake, ki jih je koračni motor moral za dano pot opraviti. Zapišemo jih v globalno spremenljivko *numberOfSteps*. Število korakov se lahko spreminja zaradi mehanskih sprememb dolžine poti, lahko pa tudi zaradi sprememb nastavitve dolžine koraka na gonilniku koračnega motorja. Po končani kalibraciji je voziček na skrajni levi točki. Inicializiramo dodatno globalno spremenljivko *numberOfStepsNow*, ki nam bo v vsakem trenutku dala informacijo o poziciji vozička in jo nastavimo na vrednost 0.

Funkcija *loop* je neskončna zanka, ki se izvaja na mikrokrmilniški ploščici. Na analognem vhodu preberemo vrednost, ki jo je uporabnik vnesel v program WinWcp in jo po enačbi (3.1) pretvorimo v korak, na katerega naj se voziček pomakne. Število korakov izračunamo po enačbi (3.10) in ga premaknemo levo, če je število pozitivno, oz. desno, za absolutno vrednost, če je število negativno.

$$stepsToDo = numberOfStepsNow - goToStep. \quad (3.10)$$

Dejanski premik vozička izvedemo z dvema metodama *goLeftStep(intstep)* in *goRightStep(intstep)*, ki delujeta zelo podobno. Razlika je v nastavitvi smeri, nato izvajanje preide v zanko *for*, ki se izvede tolikokrat, kolikor korakov smo

podali v vhodnem argumentu metode. Dejanski premik koračnega motorja za en korak izvedemo po principu, opisanem v razdelku 3.2. Znotraj zanke *for* smo implementirali še varnostni pogoj, da se premikanje ustavi, če se voziček nahaja na kateri od kritičnih točk. Pogoj je implementiran tako, da v metodi *goLeftStep(intstep)* preverja ali je stikalo na levi strani v stanju napetosti 3.3 V in v metodi *goRightStep(intstep)* ali je stikalo na desni strani v stanju napetosti 3.3 V. V primeru, da je pogoj pravilen, se ponastavi stanje trenutnega koraka, globalne spremenljivke *numberOfStepsNow* in prekine izvajanje celotne metode.

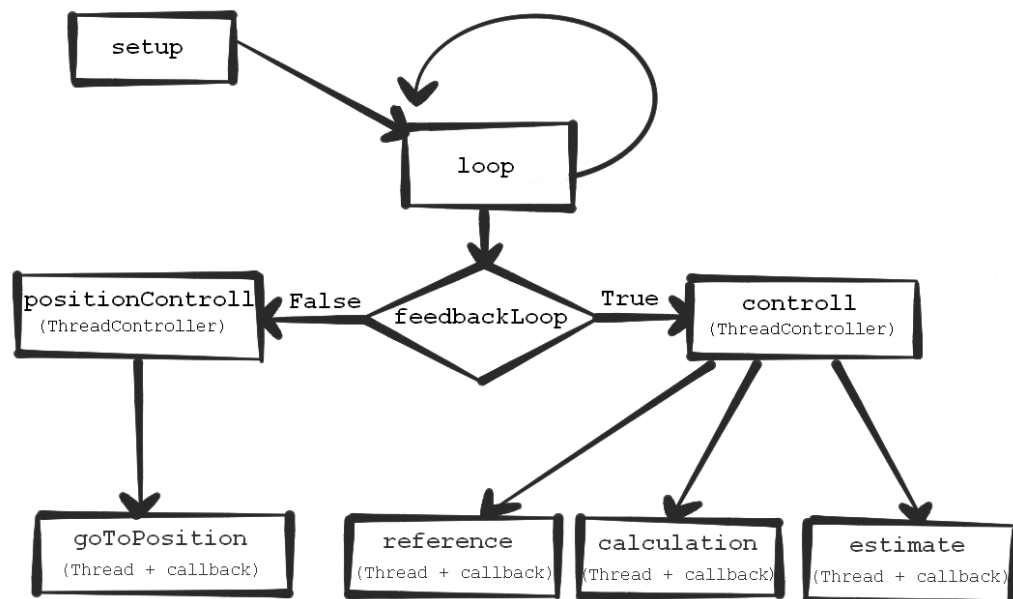
Povratna zanka

Kalibracijo sistema smo za implementacijo funkcije povratne zanke nadgradili z inicializacijo dveh globalnih spremenljivk, *minimum* in *maximum*. V obeh skrajnih legah na analognem vhodu beremo zgornje in spodnje amplitudne vrednosti električne napetosti signala. Čas, ko ima signal zgornjo amplitudno vrednost, nam določa signal, ki ga beremo iz drugega analognega vhoda. To je signal, ki ga dobimo iz fotostikala mehanske bliskovnice. V programski kodi je v obliki globalne spremenljivke *photoreceptorDelay* implementiran zamik, ki ga nastavimo na 20 ms, kolikor je reakcijska latenca fotoreceptorja, kadar z njim nadomestimo fotodiode. Dejanski vrednosti integrala v kritičnih točkah izračunamo, kot je opisano v razdelku 3.4.1 in ju zapišemo v omenjeni globalni spremenljivki.

Mikrokrmilniška ploščica ne omogoča večnitnega programiranja in v osnovi konstantno deluje znotraj zanke *loop*. Naloga je zasnovana tako, da bi program ves čas moral sprejemati vrednosti, ki jih nastavlja uporabnik in ob spremembi popraviti referenčno vrednost, hkrati pa sprejemati in preračunavati vrednosti signala, ki jih dobimo iz fotodiode oz. fotoreceptorja. Problem smo rešili z uporabo dveh C++ knjižnic *Thread* in *ThreadController*, ki simulirata delovanje večnitnega programiranja. Objekte smo inicializirali na globalni ravni, v metodi *setup* pa smo tem objektom definirali metode *callback* in časovni interval, na katerega naj se ponovi izvajanje. Arhitektura delovanja programa je prikazana na sliki 3.4.

Jedro programa ostaja v neskončni zanki funkcije *loop*. V njej se izvede ena vrstica kode; *controll.run()*. Objekt *controll* je tipa *ThreadController*.

ThreadController je razred, ki združuje več objektov oz. skupino objektov tipa *Thread*. Deluje tako, da ob izvedbi metode *run()*, na vsakem od objektov izvede



Slika 3.4: Arhitektura delovanja programa

metodo *run()* (metodi se razlikujeta med sabo, saj izvajata klic v različna razreda; *Thread.run()* in *ThreadController.run()*) v navedenem vrstnem redu. Funkcija *run()* se na objektih tipa *Thread* izvede samo takrat, kadar je to potrebno in sproži izvajanje metode *callback*, ki smo mu jo določili.

Ob inicializaciji smo določili, da se najprej sproži metoda *reference_callback()*. V njej na analognem vhodu preberemo vrednost, ki jo je uporabnik nastavil v programu WinWCP. Vrednost po enačbi (3.1) pretvorimo v korak, na katerega naj se voziček pomakne. Primerjamo izračunan korak in korak, na katerem je trenutno voziček (takoj po inicializaciji je voziček na koraku 0) in ga pomaknemo za dobljeno število korakov. Izvedemo klic metode *calculation()*, v kateri preberemo vrednosti, ki jih dobimo iz fotodiode oz. fotoreceptorja in jih preračunamo v vrednost intervala korakov, po postopku opisanem v 3.4.1. S tem smo določili referenčno vrednost, katero zapišemo v globalno spremenljivko *refStep*. Sledi izvajanje metode *calculation_callback()*. Ta ponovi postopek iz metode *calculation()*, vendar izračunane vrednosti ne zapiše v globalno spremenljivko *refStep*. Referenčna vrednost in vrednost, ki smo jo izračunali v metodi *calculation_callback()*

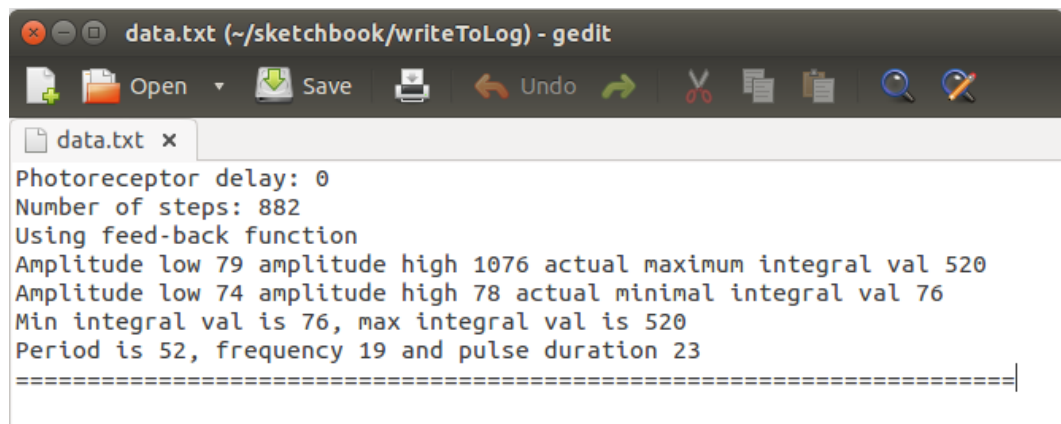
ustrezata vhodnima stanjema povratne zanke, opisane v razdelku 3.2.1. Po definiciji povratne zanke moramo ti dve stanji uskladiti. Usklajevanje se realizira v metodi *estimate_callback()*, ki je vedno izvedena takoj po preračunu vrednosti iz fotodiode oz. fotoreceptorja. Na tem mestu ugotovimo ali je usklajevanje potrebno. Če je potrebno, pomaknemo voziček za potrebno število korakov v smer, ki smo jo ugotovili na podlagi primerjave vrednosti. Izvedemo klic metode *goRightStep(intnumberOfSteps)* oz. *goLeftStep(intnumberOfSteps)*. Znotraj teh dveh metod se na vsakem koraku uporabniku posreduje informacijo o legi sivega klina. To informacijo imamo shranjeno v globalni spremenljivki *numberOfStepsNow* in jo z uporabo enačbe (3.1) pretvorimo iz intervala korakov v interval analognih vrednosti [0, 3 V]. Signal pošljemo preko izhoda DAC.

Pri branju analognih vrednosti smo opazili, da prihaja do šuma, ki sicer ni tako velik (tipično pod 1 %), da bi kvaril natančnost sistema, vendar bi vsakič znova prožil ponastavljanje globalne spremenljivke *refStep*. Ponastavljanje oz. izvajanje metode *calculation()* traja vsaj dve periodi. Predvidevali smo, da uporabnik vrednosti ne bo spreminjal prav pogosto, zato bi bilo izvajanje te metode v vsaki iteraciji zanke *loop* potratno, poleg tega pa bi občutno upočasnilo odzivnost povratne zanke. Ob inicializaciji smo nastavili interval izvajanja na 5000 ms in s tem dosegli, da se metoda ne izvaja prepogosto. Interval smo določili empirično.

Spremljanje vrednosti, ki jih dobimo iz fotodiode oz. fotoreceptorja in usklajevanje z referenčno vrednostjo je za delovanje povratne zanke ključnega pomena, zato smo interval izvajanja nastavili na 1 ms metodama *calculation_callback()* in *estimate_callback()*. Metodi se skupaj zagotovo izvajata dalj kot 1 ms, kar pomeni, da se bosta obe izvedli v vsaki iteraciji zanke *loop*. S tem vzdržujemo vrstni red izvajanja teh dveh metod. Metodi bi sicer zaradi konstantne zaporednosti lahko združili v eno, vendar smo se zaradi preglednosti programske kode in vsebinske strukture odločili, da ju ločimo.

Beleženje v dnevnik

Za lažje razhroščevanje smo dodali funkcionalnost, ki zapisuje niz znakov v tekstovno datoteko. Programu, napisanem v programskem jeziku Arduino, definiramo povezavo zaporednih vrat na 9600 bitov na sekundo z ukazom *Serial.begin(9600)*. Zaporedna vrata je izraz, ki pomeni priključek v računalniku, namenjen zapore-



```
data.txt (~/.sketchbook/writeToLog) - gedit
Open Save Undo
data.txt x
Photoreceptor delay: 0
Number of steps: 882
Using feed-back function
Amplitude low 79 amplitude high 1076 actual maximum integral val 520
Amplitude low 74 amplitude high 78 actual minimal integral val 76
Min integral val is 76, max integral val is 520
Period is 52, frequency 19 and pulse duration 23
=====
```

Slika 3.5: Primer zapisa kalibracijskih parametrov v log

dnemu prenosu podatkov med računalnikom in zunanjo napravo [23]. Za zapis niza znakov preko vrat uporabimo ukaz *Serial.println(string)*. Ob zagonu programa na mikrokrmilniški ploščici hkrati zaženemo program, napisan v programskem jeziku Processing. V njem inicializiramo nov objekt, tipa *Serial*, vzpostavimo povezavo z vrati *ACM0* in mu določimo enako hitrost, 9600 bitov na sekundo. Ob prejemu niza znakov, tega zapiše v tekstovno datoteko, ki smo jo poimenovali *data*.

Beleženje ene vrstice niza v dnevnik je zaradi prenosa podatkov preko zaporednih vrat časovno zahteven ukaz [17], v našem programu tipično 27 ms. Zato se je beleženje v jedru programa uporabljalo le za čas razvoja. V dejanskem delovanju programa se beleženje uporablja zgolj ob zagonu sistema, med kalibracijo. Čas trajanja kalibracije je lahko poljuben, saj se izvede samo enkrat. V dnevnik se zapišejo ključni parametri, ki determinirajo delovanje programa. Primer zapisa kalibracijskih parametrov je prikazano na sliki 3.5.

Rezultati

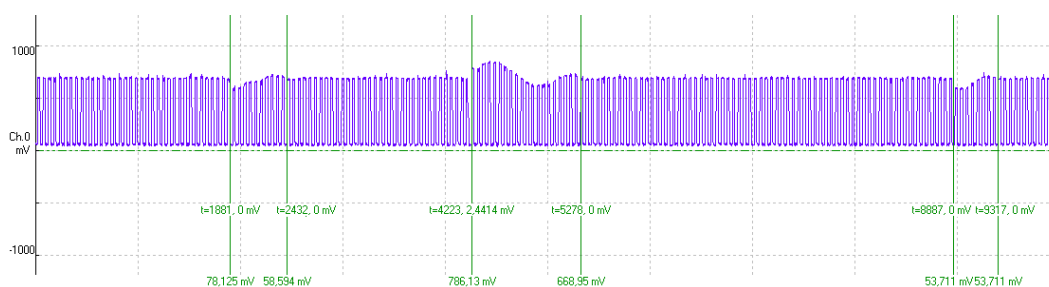
Prvi del diplomske naloge, postavitve sivega klina na fiksno pozicijo, deluje po pričakovanjih. Sistem se odziva hitro in natančno.

Sistem nadzora sivega klina v povratni zanki se odziva hitro in dovolj natančno. Iz slike 3.9 lahko razberemo, da se je sistem po obeh spremembah stanja odzval po dveh periodah in za dosego zelenega stanja v prvem primeru porabil 0.5686 s,

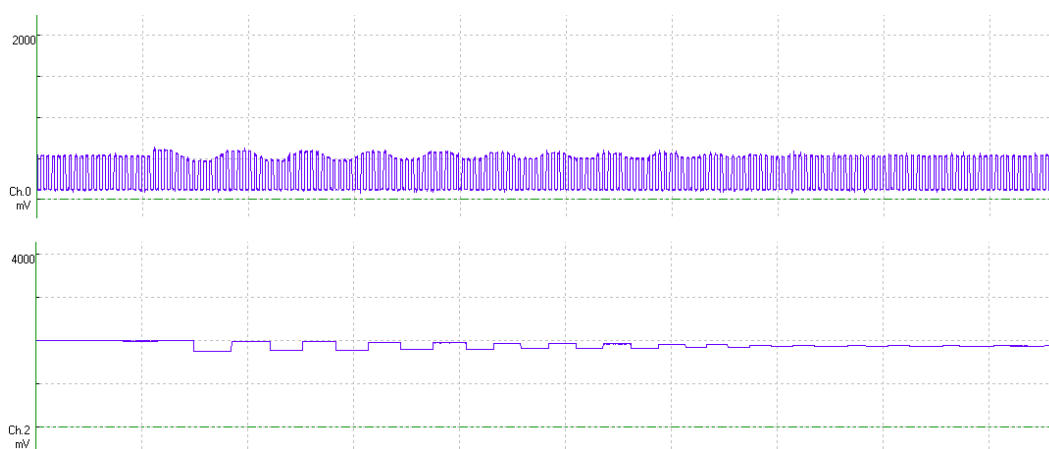
v drugem pa 0.359 s. Pri nastavljeni frekvenci 20 Hz to pomeni, da je izmerjena vrednost skonvergirala proti referenčni v 11 periodah v prvem primeru in v 7 periodah v drugem.

Za izračun vrednosti integrala beremo in povprečimo vrednosti znotraj intervala, kot je opisano v razdelku 3.4.1. V tem času sistem miruje, potem pa se popravi na izračunano vrednost (slika 2.1). Interval branja vrednosti smo nastavili na čas dveh period. Izmerjena in preračunana vrednost je bila zadovoljivo točna, saj je hitro konvergirala proti referenčni, tipično pod 0.5 s. Sistem je potreboval majhno število iteracij popravkov (primer prikazan na sliki 3.6). Sistem smo želeli pospešiti tako, da smo skrajšali čas branja vrednosti, torej čas ko sistem miruje, na interval ene periode. Ob tem se je pojavil problem, da izmerjene in izračunane vrednosti niso bile dovolj točne. Posledično je sistem potreboval veliko večje število iteracij popravkov. Čas, ki ga sistem zdaj porabi za vzpostavitev zelenega stanja, je kljub krajšemu času branja vrednosti daljši zaradi večjega števila iteracij popravkov, kot če interval podaljšamo in na ta račun dobimo bolj točne vrednosti. Počasna konvergenca k pravi vrednosti, pri nastavitvi intervala na čas ene periode, je prikazana na sliki 3.7. Ob enem se je izkazalo, da se sistem, pri povprečenju vrednosti na intervalu ene periode obnaša nestabilno. Pri istih nastavitvah in pogojih je potreboval občutno različen čas za vzpostavitev zelenega stanja.

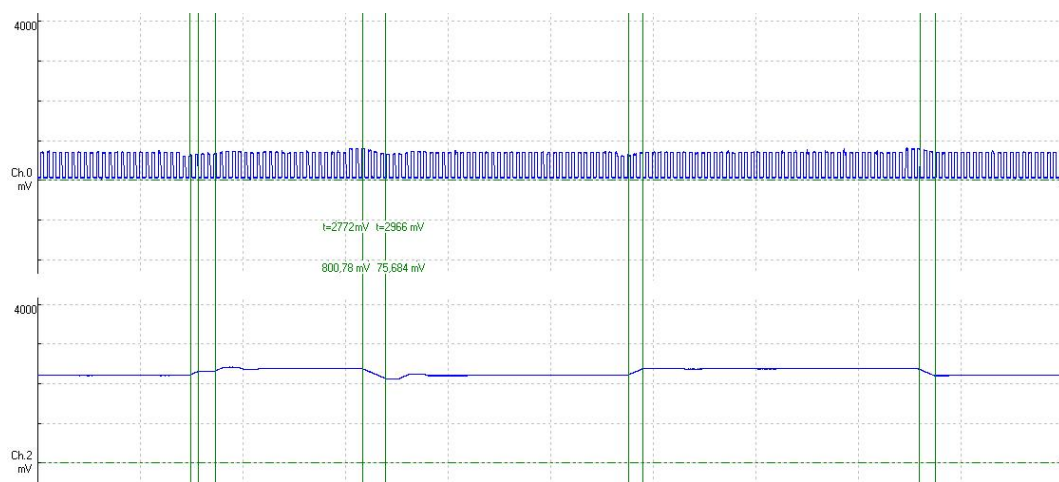
Uporabnik pri uporabi funkcije povratne zanke potrebuje informacijo za koliko je bilo potrebno popraviti sistem po zaznanem odstopanju od želene vrednosti. Informacijo posredujemo programu WinWCP s signalom preko izhoda DAC, kot je razloženo v razdelku 3.4.2. Pošiljanje informacije o poziciji sivega klina preko izhoda DAC se izvede na vsakem koraku, zaradi česar so dejanski popravki sistema in informacija o poziciji sivega klina sinhronizirani (prikazano na sliki 3.8). Prenos podatkov preko izhoda DAC je dovolj hiter, da ne upočasnjuje sistema in odzivnosti povratne zanke. Sistem je za vzpostavitev zelenega stanja še vedno potreboval okoli 0.5 s, kot je razvidno iz slike 3.9.



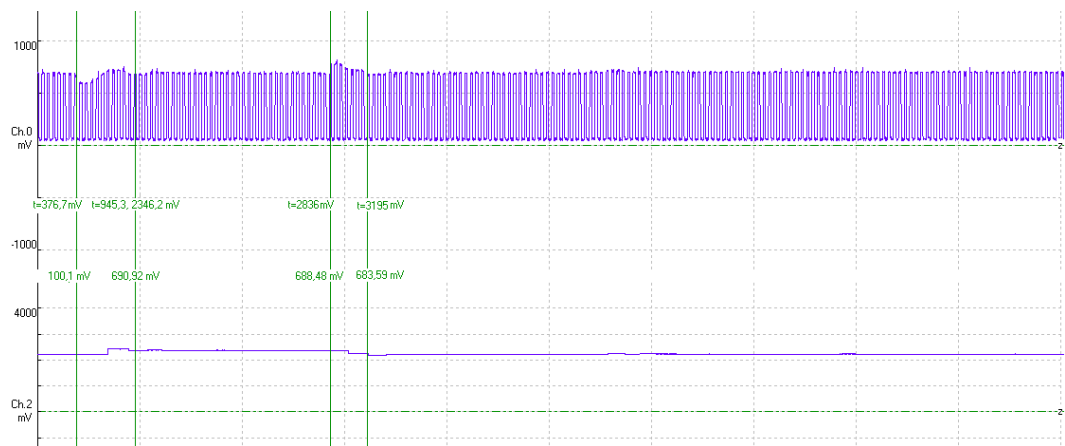
Slika 3.6: Signal iz fotodiode, zapisan v programu WCP. Vertikalni pari kurzorjev označujejo čas od začetka posnetka, zaradi napake v WCP je prikazan z enoto mV (kurzor 1: 1881,0 mV = 1881,0 ms). Prikazan je odziv sistema v povratni zanki pri spremembi stanja (ob času 1881,0 ms, 4223,2 ms in 8887,0 ms) in ob vzpostavitvi zelenega stanja (ob času 2432,0 ms, 5278,0 ms in 9317,0 ms). Interval beleženja vrednosti je bil nastavljen na čas dveh period in izpisovanje na izhod DAC izklopljeno.



Slika 3.7: Signal iz fotodiode, zapisan v programu WinWCP. Na zgornjem grafu je prikazan odziv sistema v povratni zanki pri spremembi stanja. Na spodnjem grafu je prikazan odziv sistema, ki smo ga poslali preko izhoda DAC, na 2. analogni vhod CED 1401. Odziv sistema in prikaz odziva nista sinhronizirana. Interval beleženja vrednosti je bil nastavljen na čas ene periode.



Slika 3.8: Signal iz fotodiode, zapisan v programu WinWCP. Na zgornjem grafu je prikazan odziv sistema v povratni zanki pri spremembi stanja. Na spodnjem grafu je prikazan odziv sistema, ki smo ga poslali preko izhoda DAC, na 2. analogni vhod CED 1401. Prikazan je odziv sistema v povratni zanki pri spremembi stanja (ob času 2772 ms in 2966 ms) in sinhroniziran zapis odziva sistema. Interval beleženja vrednosti je bil nastavljen na čas dveh period.



Slika 3.9: Signal iz fotodiode, zapisan v programu WinWCP. Na zgornjem grafu je prikazan odziv sistema v povratni zanki pri spremembi stanja. Na spodnjem grafu je prikazan odziv sistema, ki smo ga poslali preko izhoda DAC, na 2. analogni vhod CED 1401. Prikazan je odziv sistema v povratni zanki pri spremembi stanja (ob času 376,7 ms in 2836 ms) in ob vzpostavitvi zelenega stanja (ob času 945,3 ms in 3195 ms). Interval beleženja vrednosti je bil nastavljen na čas dveh period.

Poglavje 4

Sklepne ugotovitve

Diplomsko delo je zelo interdisciplinarno naravnano, saj pokriva več področij, kot so programiranje, matematika, elektrotehnika, fizika in biologija. To je doprineslo k pestrosti dela, hkrati pa mi je omogočilo uporabo znanja, ki sem ga pridobila tekom študija, tako na Fakulteti za računalništvo in informatiko, kot tudi na Fakulteti za matematiko in fiziko. Poleg tega sem poglobila znanje iz področij, s katerimi se bom srečevala v nadaljevanju študija.

Problem izhaja iz dejanske potrebe, zato je bil sistem v večji meri definiran, cilj jasno zastavljen in prioritete določene. Kljub temu sem pri samem razvoju imela dovolj svobode pri odločanju glede načinov, metod in postopkov, s katerimi sem opravila zastavljeno delo. Zaradi statičnosti sistema sem program razvijala na Biotehniški fakulteti, kar mi je dalo možnost interakcije z naročnikom (prof. dr. Kreft, somentor doc. dr. Belušič) in potencialnimi uporabniki (zaposleni in študentje na Biotehniški fakulteti). Zaradi tega smo že med razvojem dobili več idej, kako bomo lahko sistem v prihodnosti še dodatno nadgradili.

Pri analizi vida bi jim bila v veliko pomoč interaktiven prikaz in predobdelava podatkov, ki jih vračamo iz naše komponente. Implementacija filtrov signalov bi zmanjšala šum in pripomogla k natančnosti sistema. Sistemu bomo dodali mehanski analogni gumb, s katerim bo uporabnik nastavljal pozicijo sivega klina. To je le nekaj idej, zrelih za realizacijo.

Cilj je bil tudi implementacija uporabniškega grafičnega vmesnika, vendar je bila umestitev zaradi arhitekture fotostimulatorja komplicirana in presega obseg diplomske naloge. Na tej točki je preklapljanje med funkcijama programa in

določanje toleranc možno s spreminjanjem globalnih konstant v programski kodi. Uporabniški grafični vmesnik in dodatne funkcije, s katerimi bi lahko nadgradili sistem, se bodo implementirali ob morebitnem nadaljnjem sodelovanju.

Sistem fotostimulatorja ni bil točno definiran in postavljen pred začetkom razvoja, zato smo nekaj časa porabili za spremembe in gradnjo sistema. To je upočasnilo razvoj, hkrati pa mi je dalo vpogled v nastajanje in gradnjo celotnega produkta, kar dojemam kot kvalitetno in koristno izkušnjo.

Z razvojem mikrokrmilniške ploščice Arduino sem se srečala prvič. S predznanjem iz programiranja v različnih objektnih programskih jezikih, med drugim tudi C++, na katerem bazira programski jezik Arduino, proces učenja ni bil dolgotrajen in zahteven. Poleg tega ima celoten programski paket Arduino obsežno in kvalitetno napisano dokumentacijo.

Prednost ploščice Arduino je v ceni in preprostosti. Pokazali smo, da je implementacija rešitve s ploščico Arduino dovolj hitra in natančna, hkrati pa je poceni in nezahtevna. Zaradi splošnega in preprostega koncepta je primerna za realizacijo različnih idej v vsakodnevnem življenju. Z izdelavo programa, ki je bil cilj diplomske naloge, sem pridobila znanje, ki ga bom zagotovo koristno uporabila v prihodnje.

Literatura

- [1] A. S. French, "The linear dynamic properties of phototransduction in the fly compound eye", *The Journal of physiology*, 1980, 308.1: 385-401.
- [2] C. J. Aswell, J. H. Berlien Jr, "Apparatus and method for active integrator optical sensors", U.S. Patent No 6,031,217, 2000.
- [3] C. S. Choe, R. B. Welch, "The ventriloquist effect: Visual dominance or response bias?", *Perception & Psychophysics*, 1975, 18.1: 55-60.
- [4] D. A. Clark, R. Benichou, "Dynamical Adaptation in Photoreceptors", *PLoS computational biology*, 2013, 9.11: e1003289.
- [5] D. Gransden, P. Lonjoux., "Variable optical attenuator", U.S. Patent No. 6,404,970. 11 Jun. 2002.
- [6] G. Kopitkovas, T. Lippert, "Surface micromachining of UV transparent materials", *Thin Solid Films*, 2004, 453: 31-35.
- [7] J. G. J. Smakman, B. A. Pijpker, "An analog-digital feedback system for measuring photoreceptor properties with an equal response method", *Journal of neuroscience methods*, 1983, 8.4: 365-373.
- [8] J. Pokorny, H. Smithson, "Photostimulator allowing independent control of rods and the three cone types", *Visual Neuroscience*, 2004, 21.03: 263-267.
- [9] K. Djupsund, E. Kouvalainen, "A digital feedback controller application: a light response control system to reveal photoreceptor adaptation", *IEEE transactions on bio-medical engineering*, 1993, 40.2: 205-208.

-
- [10] M. Barr, "Pulse width modulation", *Embedded Systems Programming*, 2001, 14.10: 103-104.
 - [11] M. O. Ernst, H. H. Bühlhoff, "Merging the senses into a robust percept", *Trends in cognitive sciences*, 2004, 8.4: 162-169.
 - [12] M. Weckström, E. Kouvalainen, "Non-linearities in response properties of insect visual cells: An analysis in time and frequency domain", *Acta physiologica scandinavica*, 1988, 132.1: 103-113.
 - [13] N. Mitsunaga, "An interpreted language with debugging interface for a micro controller", In *Consumer Electronics (GCCE)*, 2012 IEEE 1st Global Conference on, pp. 115-119. IEEE, 2012.
 - [14] O. Kontovourkis, "Computational mechanisms in architectural adaptation and interaction process through human behavior physical inputs", *Architecturae et Artibus*, 2014, 6.
 - [15] P. Padmos, D. V. Norren, "The vector voltmeter as a tool to measure electroretinogram spectral sensitivity and dark adaptation", *Investigative Ophthalmology & Visual Science*, 1972, 11.9: 783-788.
 - [16] S. B. Laughlin, "The sensitivities of dragonfly photoreceptors and the voltage gain of transduction", *Journal of comparative physiology* 111.3 (1976): 221-247.
 - [17] T. W. Schubert, A. D'Ausilio, "Using Arduino microcontroller boards to measure response latencies", *Behavior research methods* 45.4 (2013): 1332-1346.
 - [18] Pulse-width modulation dostopno na:
http://en.wikipedia.org/wiki/Pulse-width_modulation
 - [19] Arduino software dostopen na:
<http://arduino.cc/en/Main/Software>
 - [20] Arduino Due dostopen na:
<http://arduino.cc/en/Main/arduinoBoardDue>
 - [21] Fotodioda dostopna na:
<http://www.islovar.org/izpisclanka.asp?id=9645>

- [22] Povratna zanka dostopna na:
<http://www.islovar.org/izpisclanka.asp?id=4930>
- [23] Zaporedna vrata dostopna na:
<http://www.islovar.org/izpisclanka.asp?id=6021>